

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería de Tecnologías y Servicios de  
Telecomunicación

TRABAJO FIN DE GRADO

**DESARROLLO DE UNA  
PLATAFORMA DE APLICACIÓN  
PARA LA MEJORA DE LAS  
HABILIDADES PSICOMOTRICES  
EN PERSONAS MAYORES**

Autor: Inmaculada Contreras de Cózar

Tutor: Alfonso Ortega de La Puente

Junio 2018



# DESARROLLO DE UNA PLATAFORMA DE APLICACIÓN PARA LA MEJORA DE LAS HABILIDADES PSICOMOTRICES EN PERSONAS MAYORES

Autor: Inmaculada Contreras de Cózar  
Tutor: Alfonso Ortega de La Puente

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Junio 2018





## Resumen

A partir de cierta edad, las personas van perdiendo movilidad y, especialmente, rapidez en general, tanto física como mentalmente. Sin embargo, se sabe que esta pérdida gradual e inevitable de habilidades psicomotrices se puede ralentizar e incluso revertir. Es por ello que este Trabajo de Fin de Grado tiene como objetivo final la creación de una plataforma que ayude a mejorar las habilidades psicomotrices en personas mayores, incluso aquellas con movilidad muy reducida.

A pesar de la existencia de ejercicios para la mejora de la movilidad y aplicaciones de juegos para la mejora de la rapidez mental, esta plataforma se ha desarrollado para unir la mejora de habilidades motrices y mentales en un ambiente lúdico, siendo así un medio completo y más efectivo para conseguir las mejoras psicomotrices deseadas.

La plataforma completa consiste en una serie de juegos desarrollados en una aplicación Android que, además, se comunica con un sistema electrónico externo. Este sistema electrónico externo, perteneciente al departamento *DTU Elektro* de la Universidad Técnica de Dinamarca donde se ha desarrollado gran parte este proyecto, consiste en unas baldosas electrónicas provistas de sensores de presión y LEDs RGB sirviendo de medio para poner en práctica los juegos desarrollados en la aplicación.

Para comenzar se ha implementado la comunicación entre sistema electrónico y aplicación mediante una serie de controladores desarrollados en C y Java, basando dicha comunicación en el uso de la tecnología de red inalámbrica de sensores *ANT*.

Después, se han diseñado y desarrollado en Java tres juegos especialmente creados para conseguir la mejoría de las habilidades psicomotrices en personas mayores. La elección de los juegos se ha llevado a cabo después de realizar consultas a personas con conocimientos del área de interés. Así mismo, se ha evaluado realmente cuál es el impacto en la mejora de la psicomotricidad en personas mayores al hacer uso de esta plataforma en una residencia de ancianos.

Finalmente, tras la recogida de datos, éstos se han analizado para observar cuantitativamente la mejora producida para así realizar una serie de conclusiones que sirvan como base para un trabajo futuro posterior.

## Palabras Clave

Aplicación, Android, Psicomotricidad, Personas mayores, Juegos electrónicos



## Abstract

When a certain age is reached, people start losing mobility and, specially, the ability to be agile, both physically and mentally. However, it is known that this gradual and inevitable loss of abilities can be slowed down and even regained up to some extent if people work on it. This is the main reason why this end of bachelor thesis has as objective the creation of a platform that will enable the elder population to improve physical and mental abilities, even those with very reduced mobility.

Despite the existence of exercises to improve mobility and game apps to improve mental agility. This platform has been developed to join the improvement of physical and mental abilities in a fun atmosphere, being then the most effective way to achieve the improvement desired.

The complete platform consists on a variety of games developed in an Android application that communicates with an external electronic system. This external electronic system belongs to the *DTU Elektro* department from the Technical University of Denmark, where most of this project has been developed, and consists of several electronic tiles with pressure sensors and RGB LEDs integrated.

First, the communication between electronic tiles and Android application has been developed in C and Java, with the use of wireless sensor network technology *ANT*.

Afterwards, three games have been designed and developed in Java, specially created to achieve an improvement of physical and mental abilities in the elder population. These games have been chosen after taking advice from people with knowledge in elder care. In addition, the real impact of this platform in the improvement of physical and mental abilities has been evaluated with practice sessions in a day care facility for elder people.

Finally, after the data recollection, it has been analyzed to observe in a quantitative way the improvement achieved in order to make several conclusions that can be used as the foundation for future work.

## Key words

App, Android, Psicomotricity, Elderly, Electronic games



# Agradecimientos

En primer lugar quería agradecer la confianza e ilusión mostrada por parte de mi tutor Alfonso en este proyecto, siempre estando disponible a cualquier hora para resolver hasta las dudas más tontas que me surgían.

Este proyecto, desarrollado con muchísimo entusiasmo, no podría haber salido adelante sin las herramientas y ayuda que me ofreció el departamento de *DTU Elektro* de la Universidad Técnica de Dinamarca, al cuál le estoy muy agradecida por la oportunidad que me ofrecieron.

También agradecer la participación en este proyecto a la residencia de mayores *Peder Lykke Centret* que con tanta energía ayudaron a probar esta aplicación en un entorno real, y gracias a mi vikingo que hacía las veces de traductor cuando mi vocabulario en danés se quedaba corto.

Finalmente dar las gracias a las personas que, tanto en los días buenos como en los malos, siempre estuvieron y están a mi lado para animarme y empujarme a no rendirme. En especial a mis padres, muchísimas gracias.



# Índice general

<b>Índice de Figuras</b>	<b>x</b>
<b>Índice de Tablas</b>	<b>xiii</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación del proyecto . . . . .	1
1.2. Objetivos y enfoque . . . . .	2
1.3. Organización de la memoria . . . . .	2
<b>2. Estado del arte</b>	<b>5</b>
2.1. Introducción . . . . .	5
2.2. Psicomotricidad y herramientas para su mejora . . . . .	5
2.3. Conclusión . . . . .	6
<b>3. Definición del sistema</b>	<b>9</b>
3.1. Metodología . . . . .	9
3.2. Herramientas usadas . . . . .	11
<b>4. Diseño</b>	<b>13</b>
<b>5. Desarrollo</b>	<b>17</b>
5.1. Controlador del sistema externo . . . . .	17
5.2. Aplicación Android . . . . .	20
<b>6. Experimentos realizados y resultados</b>	<b>27</b>
6.1. Experimentos realizados . . . . .	27
6.2. Resultados y análisis . . . . .	28
<b>7. Conclusiones y trabajo futuro</b>	<b>37</b>
<b>Bibliografía</b>	<b>39</b>
<b>A. Controlador en Java</b>	<b>43</b>





# Índice de Figuras

2.1. Logo de Stimulus . . . . .	6
2.2. Logo de Lumosity . . . . .	6
3.1. Diagrama de la metodología ágil seguida . . . . .	9
3.2. Logo de Keil $\mu$ Vision 5 . . . . .	11
3.3. Logo de Android Studio . . . . .	11
3.4. Logo de <i>ANT</i> . . . . .	11
4.1. Arquitectura de la plataforma . . . . .	14
5.1. Sistema externo . . . . .	17
a. Baldosa electrónica . . . . .	17
b. Baldosas electrónicas en su base de carga . . . . .	17
5.2. Placa y conector . . . . .	18
a. Placa integrada en las baldosas . . . . .	18
b. Conector <i>J-Link</i> . . . . .	18
5.3. Parte del controlador en <i>C</i> . . . . .	18
a. Identificación en cada baldosa . . . . .	18
b. Constantes del controlador . . . . .	18
5.4. Sistema electrónico funcionando . . . . .	19
a. Baldosas apagadas . . . . .	19
b. Baldosas iluminadas realizando la función <i>SPIN_ALL</i> . . . . .	19
5.5. Colores asignados a las baldosas . . . . .	19
5.6. Secuencia de ejemplo del juego <i>Simon Says</i> . . . . .	21
a. Baldosas apagadas . . . . .	21
b. Colores asignados . . . . .	21
c. Primer elemento . . . . .	21
d. Segundo elemento . . . . .	21
e. Tercer elemento . . . . .	21
f. Cuarto elemento . . . . .	21
g. Fin de la secuencia . . . . .	21

---

5.7. Secuencia de ejemplo del juego <i>Odd</i> . . . . .	21
a. Baldosas apagadas . . . . .	21
b. Colores asignados . . . . .	21
c. Primer elemento . . . . .	21
d. Segundo elemento . . . . .	21
e. Tercer elemento . . . . .	21
f. Cuarto elemento . . . . .	21
g. Quinto elemento . . . . .	21
h. Fin de la secuencia . . . . .	21
5.8. Secuencia de ejemplo del juego <i>Hit First</i> . . . . .	22
a. Baldosas apagadas . . . . .	22
b. Colores asignados . . . . .	22
c. Único elemento . . . . .	22
d. Fin de la secuencia . . . . .	22
5.9. Menú principal de la aplicación . . . . .	23
5.10. Interfaz de usuario de los juegos . . . . .	24
a. Pantalla del juego <i>Odd</i> . . . . .	24
b. Pantalla del juego <i>Hit First</i> . . . . .	24
5.11. Ejemplo de datos de tiempos . . . . .	24
6.1. Usuarios utilizando la plataforma . . . . .	28
6.2. Gráfica con los resultados de cada usuario del juego <i>Simon Says</i> . . . . .	30
6.3. Gráfica con los resultados de cada usuario del juego <i>Odd</i> . . . . .	31
6.4. Gráfica con los resultados de cada usuario del juego <i>Hit First</i> . . . . .	31
6.5. Gráfica con los resultados promedio de tiempo y porcentaje de fallos de las tres sesiones del juego <i>Simon Says</i> . . . . .	32
6.6. Gráfica con los resultados promedio de tiempo y porcentaje de fallos de las tres sesiones del juego <i>Odd</i> . . . . .	33
6.7. Gráfica con los resultados promedio de tiempo de las tres sesiones del juego <i>Hit First</i> . . . . .	34
6.8. Gráfica con los resultados promedio de tiempo de las diez sesiones del juego <i>Simon Says</i> obtenidos por los dos equipos . . . . .	34

## Índice de Tablas

6.1. Resultados de tiempos del juego <i>Simon Says</i> . . . . .	28
6.2. Resultados de tiempos del juego <i>Odd</i> . . . . .	29
6.3. Resultados de tiempos del juego <i>Hit First</i> . . . . .	29
6.4. Resultados de tiempos de las primeras cinco sesiones del juego <i>Hit First</i> realizado por equipos . . . . .	30
6.5. Resultados de tiempos de las cinco últimas sesiones del juego <i>Hit First</i> realizado por equipos . . . . .	30



# 1

## Introducción

### 1.1. Motivación del proyecto

---

Las personas van, inevitablemente, perdiendo habilidades motoras y rapidez mental con el paso de los años. Sin embargo, se puede ralentizar el proceso e incluso revertirlo si se practican regularmente ejercicios diseñados específicamente para la mejora de habilidades psicomotrices.

Un ejemplo de cómo la mejora de habilidades motoras influye directamente en la calidad de vida de la persona es que ayuda a la prevención de caídas, ya que se fortalecen los músculos y se gana equilibrio. Por otro lado, la mejora de la rapidez mental ayuda a que la persona se sienta más activa y capaz.

El concepto de juego se describe como: "*Actividad libre y voluntaria que hacemos con el único propósito de divertirnos. Es cuando jugamos que experimentamos una sensación de felicidad y libertad para expresarnos*" [1].

En este Trabajo de Fin de Grado se une el concepto de juego explicado anteriormente junto con el objetivo final de este proyecto. Motivado en base a que, pese a que quizás, desde un punto de vista médico sea suficiente la realización de ejercicios como los mencionados anteriormente para obtener resultados de mejora de las habilidades, nuestra hipótesis es que si estos ejercicios se presentan de manera lúdica y amena, los usuarios estarán más motivados a participar y a esforzarse en mejorar su rendimiento.

Lo mencionado en el párrafo anterior se espera que produzca mejores resultados y con mayor rapidez debido a que cuando jugamos tendemos a centrarnos en el juego, a intentar superarnos a nosotros mismos o a otros si estamos compitiendo. En definitiva, divirtiéndonos nos olvidamos de si la tarea parecía o no apetecible en un principio.

Otra motivación para este trabajo de fin de grado es que, aunque ya existan en el mercado juegos para mejorar habilidades cognitivas, no suelen tener una plataforma que incluya una mejora de las habilidades motoras generales. Además de no ser, quizás, intuitiva para el usuario y/o no se puedan cuantificar fácilmente las mejoras que se pudieran conseguir.

Por tanto, esta plataforma une estas motivaciones transformadas en objetivos para conseguir una mejora de la movilidad y de la rapidez mental ofreciendo ejercicios en forma de juegos mediante una aplicación con interfaz de usuario minimalista que hace uso de un sistema electrónico

---

externo, convirtiéndolo en una plataforma completa. De esta manera, mejorando las habilidades psicomotrices de las personas mayores se ayuda al aumento de su calidad de vida y felicidad.

## 1.2. Objetivos y enfoque

---

El objetivo principal de este proyecto es conseguir crear una plataforma que realmente mejore las habilidades psicomotrices en personas de la tercera edad de una forma entretenida para así ganar mayor motivación y resultados. Además, se ha procurado que la plataforma sea flexible y adaptable. Capaz de adecuarse a cualquier lugar y necesidades específicas de cada persona.

Esta plataforma tiene dos partes bien diferenciadas. Una parte es la aplicación Android que contiene la lógica de los juegos y, además, sirve como interfaz de usuario para la comunicación con las baldosas electrónicas. La otra parte es el sistema externo electrónico mencionado: las baldosas.

Se han centrado los esfuerzos, sobre todo, en llevar a cabo una lógica de juegos que se ajusta a los objetivos deseados, después de las sugerencias propuestas por personas con conocimientos médicos. Además, se ha desarrollado una interfaz de usuario lo más minimalista posible, que sirviera como comunicación entre usuario y baldosas electrónicas, así como la captación de datos de los participantes para luego poder analizarlos y ver cuantitativamente la mejoría producida.

Así mismo, al utilizar un sistema externo como son las baldosas provistas de sensores de presión y LEDs RGB lo que se pretende conseguir es que el usuario haga un mayor uso de los músculos del cuerpo. Esto es porque es necesario hacer mayor fuerza, y se tiene la posibilidad de usar tanto los brazos como las piernas. En cambio, si sólo se utilizara un dispositivo móvil, como una *tablet*, simplemente sería necesario utilizar las manos, lo cual limita mucho la mejora potencial de la motricidad.

## 1.3. Organización de la memoria

---

Esta memoria de trabajo de fin de grado se organiza en seis grandes secciones, además de esta primera sección introductoria:

- Estado del arte
- Definición del sistema
- Diseño
- Desarrollo
- Experimentos realizados y resultados
- Conclusiones y trabajo futuro

La sección donde se describe el estado del arte comienza con una breve explicación de qué es la psicomotricidad y de qué manera pueden mejorarse las habilidades relacionadas con la psicomotricidad. Así mismo, se describen otros sistemas investigados cuyo propósito sería también el de la mejora de las habilidades psicomotrices. De esta manera, a modo de conclusión del estado del arte pueden observarse carencias en estos sistemas, relativamente similares, ya existentes que motivaron el diseño actual de este proyecto.

---

Las siguientes tres secciones, *definición del sistema*, *diseño y desarrollo* son, sin duda alguna, el núcleo de esta memoria.

En la sección titulada *definición del sistema* se detalla la metodología seguida para llevar a cabo el diseño y desarrollo de esta plataforma, junto con una breve enumeración de las tareas propuestas que se siguieron para su desarrollo, y una mención a las herramientas utilizadas.

En la siguiente sección se muestra cómo se ha diseñado este proyecto, explicando en detalle los objetivos buscados y de qué manera el diseño elegido ayuda a cumplir con estos objetivos. Así mismo, se explican las funcionalidades finales con las que cuenta la plataforma, listando explícitamente los requisitos funcionales y no funcionales más relevantes del sistema.

Seguidamente se precisa cómo se ha desarrollado e implementado este proyecto. En esta sección se explica de forma exhaustiva el funcionamiento final de la plataforma, así como de qué manera se ha implementado cada parte importante, tanto del *back-end* [2] como del *front-end* [2].

La sección titulada *experimentos realizados y resultados* es, como bien indica su nombre, donde se analizan los resultados recolectados de los experimentos realizados para ilustrar de manera cuantitativa la efectividad de la plataforma.

Finalmente se elaboran una serie de conclusiones y posibles propuestas para un trabajo futuro en el que continuar y mejorar la plataforma.

Esta memoria de Trabajo de Fin de Grado incluye, además, un índice de figuras y tablas al comienzo, y una lista de referencias de las que se hace uso al final. Así mismo, al final del documento se encuentra un anexo con el código fuente del controlador desarrollado en Java.





# 2

## Estado del arte

### 2.1. Introducción

---

Desde hace unos años ha crecido el interés entorno a maneras efectivas de mejorar habilidades físicas y mentales, de las que dependen, por ejemplo, los músculos y la memoria. En especial, aquellas de las personas mayores, ya que a medida que pasan los años la pérdida de estas habilidades motoras y de agilidad mental se hace más evidente y limita a la persona en su día a día.

En esta sección se describe qué es exactamente la psicomotricidad y se analizan las herramientas más destacadas que ayudan a mejorar este aspecto en la tercera edad.

### 2.2. Psicomotricidad y herramientas para su mejora

---

La psicomotricidad [3] es la interacción entre el movimiento y la agilidad mental de una persona, englobando numerosas habilidades [4] indispensables para el ser humano, como son el equilibrio y los reflejos. El desarrollo o deterioro de esta disciplina afecta en la calidad de vida de la persona y de cómo se relaciona con el mundo a su alrededor. Es por ello que se recomienda realizar ejercicios [5] que, de manera efectiva, ayuden a recuperar parte de estas habilidades motoras y cognitivas.

Se han seleccionado dos aplicaciones para destacar en esta memoria de trabajo de fin de grado que se describen a continuación.

#### *Stimulus*

Stimulus es una aplicación creada en España para la estimulación y rehabilitación cognitiva [6]. Esta aplicación contiene numerosos ejercicios clasificados en distintas áreas funcionales como pueden ser la memoria y el razonamiento, pasando por el lenguaje y funciones visio-motoras.

---

Está diseñada para usarse en un entorno médico, aunque cuenta con una versión doméstica. Esto es porque los usuarios a los que está enfocada la aplicación son, además de personas de la tercera edad, personas que pudieran tener alguna enfermedad neurodegenerativa.

Esta aplicación, aun siendo de pago en su versión completa, cuenta con una versión gratis dotada con un número más limitado de ejercicios.



Figura 2.1: Logo de Stimulus

### *Lumosity*

Lumosity es una aplicación estadounidense con versión en español, similar a la anterior ya que es una aplicación para el entrenamiento cognitivo [7]. Cuenta con características adicionales parecidas a Stimulus, como puede ser la recogida de datos de tiempos y aciertos, así como un análisis de ellos.

Stimulus, sin embargo, está dotado de mayor número de ejercicios que Lumosity, lo cual le concede ventaja sobre este. Además, Lumosity es una aplicación enfocada a todo tipo de personas convirtiéndolo en una aplicación menos especializada.

Lumosity es, al igual que Stimulus, de pago a pesar de contar con una versión muy limitada gratuita. En general en las dos aplicaciones, todas las características adicionales que son de interés como pueden ser la recogida de datos del rendimiento del usuario sólo están disponibles en la versión completa no gratuita.



Figura 2.2: Logo de Lumosity

## **2.3. Conclusión**

---

Tras el estudio del estado del arte, se ha visto que existen actualmente numerosos ejercicios y aplicaciones de mejora de las habilidades de interés. Sin embargo, presentan carencias que este Trabajo de Fin de Grado tiene como objetivo solventar.

La mayoría de los sistemas más modernos se basan únicamente en una aplicación online, diseñada para usarse en una *tablet*, lo cual no permite mejorar las habilidades motoras generales salvo las de la mano.

Estas características carentes en las herramientas anteriores se han tenido en cuenta para el diseño de esta plataforma, creando así un sistema completo que ayude a mejorar las habilidades psicomotrices en personas de la tercera edad de la forma más amena posible.





# 3

## Definición del sistema

### 3.1. Metodología

---

La metodología que se ha elegido el desarrollo de este proyecto es una metodología ágil de manera iterativa [8]. Se ha seguido esta metodología ya que se consideró que una metodología más convencional, como puede ser un ciclo de vida en cascada, no se ajustaba a la manera en que esta plataforma se había ideado.

Seguir una metodología ágil permitió tener mayor flexibilidad y eficacia a la hora de desarrollar la plataforma. Además, esta metodología consigue minimizar riesgos ya que no es necesario esperar a tener una versión final del proyecto para realizar mejoras y subsanar fallos.

Se realizaron mayor número de versiones de la aplicación y de la lógica de los juegos en ella, y con ello numerosas pruebas de las distintas partes. De esta forma, el desarrollo de la plataforma fue más interactivo ya que se fueron haciendo pequeños ajustes y mejoras de forma rápida a medida que se recibían comentarios de las varias entregas del proyecto. En una sección más adelante se explica en detalle el desarrollo de esta plataforma.

En la siguiente figura puede verse, a grandes rasgos, las distintas fases seguidas en este proyecto mediante una metodología ágil.

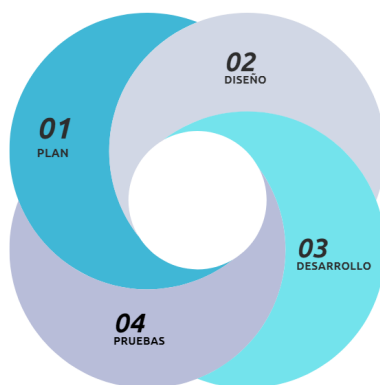


Figura 3.1: Diagrama de la metodología ágil seguida

---

A continuación se listan las tareas que componen cada una de las fases ilustradas en la figura anterior:

### **PLAN**

1. Análisis del estado del arte.
2. Estudio de ejercicios psicomotrices.
3. Especificación de objetivos y funcionalidades.

### **DISEÑO**

1. Diseño del sistema completo.
2. Diseño preciso del back-end.
3. Diseño preciso del front-end.

### **DESARROLLO**

1. Comunicación de las baldosas via ANT.
2. Implementación de funciones de las baldosas en la aplicación
3. Implementación de la lógica de juegos
4. Implementación de la interfaz de usuario

### **PRUEBAS**

1. Ideación de experimentos.
2. Desarrollo de pruebas y recopilación de datos.
3. Análisis de resultados.
4. Evaluación de la plataforma.

---

## 3.2. Herramientas usadas

---

En este apartado se describen las herramientas que se han usado para el desarrollo del proyecto, así como la tecnología utilizada para comunicar el sistema externo electrónico con la aplicación.

### *Keil $\mu$ Vision 5 MDK*

Keil  $\mu$ Vision 5 MDK [9] es un entorno de desarrollo para microcontroladores *ARM* [10]. Esta herramienta es la utilizada para cargar el *firmware* proporcionado por *DTU Elektro* en las baldosas, así como implementar en *C* [11] las funciones necesarias para la comunicación.



Figura 3.2: Logo de Keil  $\mu$ Vision 5

### *Android Studio*



Figura 3.3: Logo de Android Studio

Android Studio [12] es un entorno de desarrollo integrado que se ha utilizado para el desarrollo de la aplicación móvil. El back-end se ha implementado en *Java* [13] y la interfaz de usuario en *XML* [14].

### *Tecnología inalámbrica ANT*

La tecnología de red inalámbrica de sensores *ANT* [15] define un protocolo de comunicaciones que permite operar en la banda de frecuencia ISM [16] de 2.4 GHz. Es una tecnología similar a la de *Bluetooth* [17] pero orientada hacia el uso con sensores. Es esta tecnología la que se ha utilizado para comunicar las baldosas electrónicas con la aplicación móvil.



Figura 3.4: Logo de *ANT*





# 4

## Diseño

Al diseñar esta plataforma de mejora de habilidades psicomotrices en personas mayores, se ha tenido en cuenta lo analizado en el estado del arte, así como las recomendaciones de profesionales del sector de la salud.

Acerca de las baldosas, se han elegido cuatro colores bien diferenciados para asignar a cada una de las baldosas. Así mismo, se decidió que la cantidad de cuatro baldosas es un número que permite que los juegos tengan un nivel de complejidad adecuado sin llegar a ser demasiado difícil. Sin embargo, la forma en que se ha desarrollado el back-end de los juegos permite fácilmente aumentar o disminuir la cantidad de baldosas utilizadas, cambiando así el nivel de dificultad de los juegos para ajustarlo a la necesidad de cada persona.

Se ha querido dar gran importancia a conseguir motivar a los usuarios ya que, como se definió en los objetivos, se pretende que quien realice los ejercicios entre en la dinámica del juego y se divierta, consiguiendo mejorar sus habilidades motoras y rapidez mental sin que se dé cuenta. Una pequeña competitividad sana contra ellos mismos u otros usuarios se añade haciendo posible ganar puntos si se realiza la actividad correctamente generando motivación extra.

A continuación se listan los requisitos funcionales y no funcionales más relevantes del sistema.

### Requisitos Funcionales

- Comunicar las baldosas electrónicas con la aplicación Android
- Captar la señal de presión emitida por la baldosa).
- Elección de juegos en el menú principal.
- Comenzar el ejercicio pulsando un sólo botón.
- Registrar datos de tiempo utilizado y aciertos del usuario.

---

## Requisitos no funcionales

- Posibilidad de ampliar número de baldosas conectadas.
- Comunicación inalámbrica.
- Interfaz minimalista.
- Inclusión de explicación de cada juego.
- Imagen de disposición de las baldosas.
- Visualización de los resultados.

En definitiva, se pretende que la forma realizar los ejercicios sea simple:

1. Elegir el juego.
2. Colocar las baldosas siguiendo la disposición mostrada en la aplicación.
3. Pulsa *Start* para inicializar las baldosas.
4. Pulsa *Play* y comienza a jugar.

De esta forma, y como puede verse en la imagen a continuación, la arquitectura de la plataforma sería la siguiente:

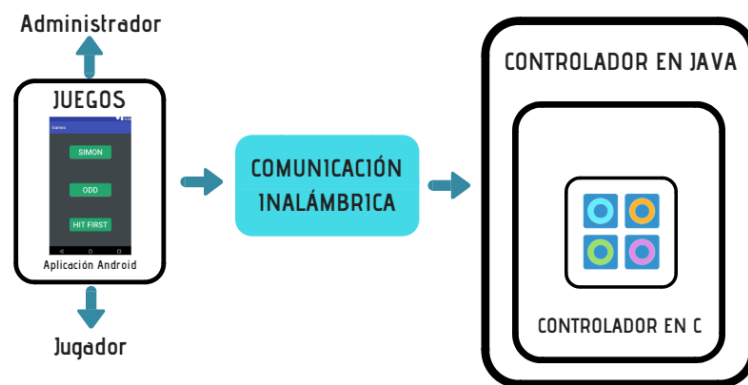


Figura 4.1: Arquitectura de la plataforma

Se diseñaron siete juegos, además de posibles variantes de cada uno de ellos, de los cuales finalmente se implementaron tres en esta plataforma. Se eligieron los juegos llamados *Simon Says*, *Odd* y *Hit First*, cuya lógica se explica en detalle en la siguiente sección.

Sin embargo, en futuras mejoras de la plataforma se pueden desarrollar las variaciones pensadas, así como nuevos ejercicios. Algunos ejemplos podrían ser la de incluir una baldosa que no se debiera pulsar en la secuencia del juego *Simon Says* o diseñar juegos que incluyan más sonidos y operaciones matemáticas sencillas.

Aprovechando la modularidad de las baldosas se jugó con la idea de poder diseñar, manualmente cambiando su posición, actividades grupales para la mejora de la psicomotricidad, que finalmente sí se pudieron implementar como puede verse en la sección 6. En general, este aspecto del sistema electrónico lo dota de flexibilidad, permitiendo al terapeuta diseñar y crear

---

numerosos tipos de ejercicios para la mejora de la movilidad y la agilidad mental, incluso en el momento de jugar.

Además, se intentó que esta plataforma fuera adaptativa. Es decir, que fuera adaptando el nivel de dificultad de los juegos en mayor o menor medida según las necesidades del usuario, ya que de esta manera se ayuda a conseguir mejoras de mayor calidad y más rápidamente.

En un principio se diseñó para implementar este aspecto de manera automática según los datos que se fueran recopilando. Sin embargo, finalmente se optó por una adaptación manual, ya que se consideró que era mejor opción dar mayor control a la persona responsable de utilizar la aplicación con fines médicos en una primera versión de esta plataforma.

Acerca de capturar datos de tiempo y aciertos de los usuarios al jugar, se diseñó en un principio la realización de una base de datos completa, para almacenar los resultados obtenidos incluso después de finalizar la sesión. La idea de recolección de datos tiene como objetivo no solo cuantificar la mejora de la psicomotricidad producida, sino también potencialmente poder utilizar esos datos para adaptar la dificultad de los juegos según el usuario.

Sin embargo, por falta de tiempo, finalmente se implementó en la aplicación final de la plataforma una recogida de datos de tiempo y aciertos que se reinicia en cada sesión, siendo necesario realizar una captura de pantalla para poder conservar los datos para analizarlos posteriormente.



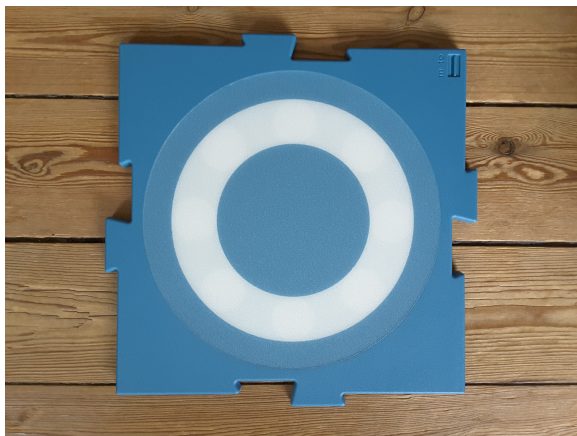
# 5

## Desarrollo

### 5.1. Controlador del sistema externo

---

Para comenzar esta sección, se muestran unas imágenes de la apariencia del sistema externo, las baldosas electrónicas. Éstas están provistas de un sensor de presión en el centro, y ocho LEDs RGB [18] alrededor de éste.



(a) Baldosa electrónica



(b) Baldosas electrónicas en su base de carga

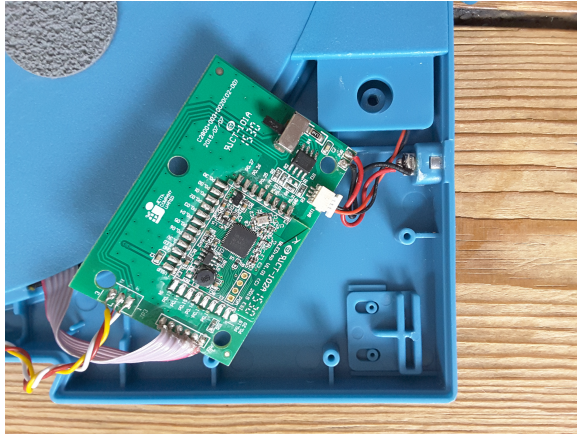
Figura 5.1: Sistema externo

Se comenzó implementando en  $C$  el controlador del sistema externo, de la manera que se describe a continuación, mediante la herramienta mencionada anteriormente: Keil  $\mu$ Vision 5 MDK.

Se instaló el *firmware* básico, dado junto con las baldosas, en la placa integrada en ellas. Seguidamente se modificaron ficheros para crear funciones necesarias para la plataforma, y se

implementó la función necesaria para la comunicación con el dispositivo móvil mediante la aplicación que se desarrolló más adelante.

En las figuras 5.2a y 5.2b puede verse la placa integrada en las baldosas (Nordic Semiconductor *nRF51422* [19]) y la conexión utilizada para instalar el controlador (*J-Link* [20]).



(a) Placa integrada en las baldosas



(b) Conector *J-Link*

Figura 5.2: Placa y conector

Primero, cada baldosa se configuró con una identificación única para que la aplicación pueda identificarlas como puede verse en la porción de código 5.3a, junto con las constantes más importantes del controlador que se muestran en la figura 5.3b. Después, se implementó la tecnología inalámbrica *ANT* dentro del controlador mediante una serie de funciones y se configuró la red en la forma de *maestro* y *esclavo* [21], siendo el dispositivo móvil con la aplicación Android el *maestro* y las baldosas los *esclavos*.

```
1 // Ant configuration IDs
2 #define MODULE_ID 3
3
4 #define MASTER_ID 0x53
```

(a) Identificación en cada baldosa

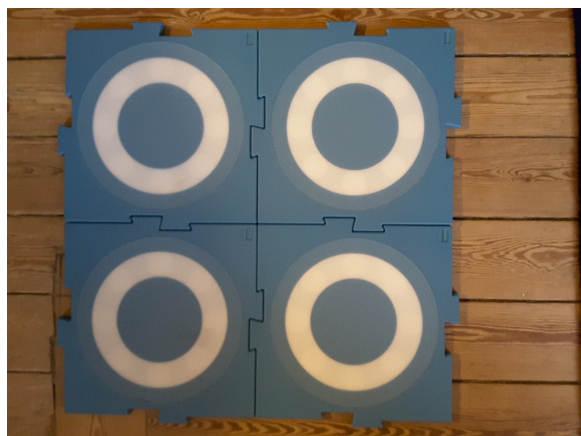
```
11 // Command codes
12 #define CMD_IDLE 0x08
13 #define CMD_SET_COLOR 0x02
14 #define CMD_FUNC_SPIN_ALL 0x03
15 #define CMD_FUNC_COUNT_UP 0x04
16 #define CMD_FUNC_COUNT_DOWN 0x05
17 #define CMD_FUNC_SPIN_INDIVIDUAL 0x06
18 #define CMD_FUNC_COUNT_UP_FADE 0x07
19 #define CMD_FUNC_RECORDING 0x09
20 #define CMD_SET_ALL 0x10
21 #define CMD_SET_OFF_ALL 0x01
22
23 // Return values
24 #define EVENT_PRESS 0x01
25 #define EVENT_RELEASE 0x09
26
27 // Color codes
28 #define LED_NUM_COLORS 9
29 #define LED_COLOR_OFF 0
30 #define LED_COLOR_RED 1
31 #define LED_COLOR_BLUE 2
32 #define LED_COLOR_GREEN 3
33 #define LED_COLOR_INDIGO 4
34 #define LED_COLOR_ORANGE 5
35 #define LED_COLOR_WHITE 6
36 #define LED_COLOR_VIOLET 7
37 #define LED_COLOR_CYAN 8
```

(b) Constantes del controlador

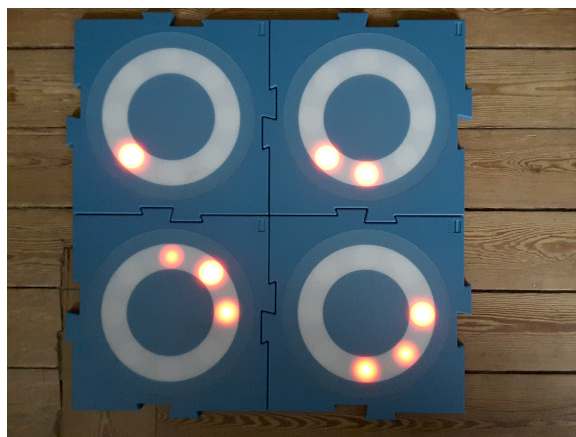
Figura 5.3: Parte del controlador en *C*



En las figuras 5.4a y 5.4b pueden verse un grupo de cuatro baldosas apagadas y ejecutando una función, respectivamente.



(a) Baldosas apagadas



(b) Baldosas iluminadas realizando la función *SPIN\_ALL*

Figura 5.4: Sistema electrónico funcionando

Así mismo, en la figura 5.5 se ven los cuatro colores elegidos de los LEDs RGB para luego ser utilizados en los juegos de la aplicación Android.



Figura 5.5: Colores asignados a las baldosas

Tras la implementación de varias funciones en las baldosas para familiarizarse con el proceso de desarrollar el controlador se vio que era más eficiente implementar la mayor parte del controlador dentro de la aplicación Android. Por ello, se desarrolló la clase de Java llamada *Tile-Commander*, cuyo código se adjunta en el anexo A que se encuentra al final de este documento.

Esta clase organiza, simplifica y unifica el proceso de controlar el sistema externo, contando con funciones que modifican los colores y la forma en que se encienden los LEDs RGB, así como funciones necesarias para la comunicación con el sensor de presión. De esta manera sólo es necesario asignar a cada baldosa nueva una identificación y asegurarse de que todas comparten el mismo canal maestro (ver figura 5.3a).

---

En concreto, en esta plataforma se configuraron ocho baldosas debido a que, aunque para los juegos desarrollados que se describirán más adelante sólo se necesitan cuatro baldosas para practicar el ejercicio de manera individual, se implementó en la aplicación el código necesario para que también fuera posible jugar uno contra uno o en equipos sin necesidad de modificar nada a la hora de realizar los experimentos.

## 5.2. Aplicación Android

---

En esta sección se explica en detalle cada ejercicio desarrollado para mejorar las habilidades psicomotrices así como la interfaz de usuario implementada en la aplicación Android.

*Android Studio* fue la herramienta de desarrollo [22] elegida para implementar en Java la lógica de los tres juegos de la aplicación Android y su interfáz de usuario.

La aplicación sigue un funcionamiento general básico: Se comunica con las baldosas mediante la tecnología inalámbrica *ANT* controlada por la clase *TileCommander*, como se ha explicado anteriormente, para así obtener las señales de presión y no presión de las baldosas y en base a esos datos encender y apagar los LEDs RGB integrados en las baldosas siguiendo la lógica de cada juego.

La aplicación de la plataforma, tal y como se ha diseñado, está formada por tres juegos, independientes entre sí cuya lógica se explica en los párrafos siguientes. Además, como puede verse más adelante en 5.10b tanto la interfaz de usuario, como la lógica de los juegos se ha implementado para que pueda jugarse de manera individual o en equipos enfrentados.

### *Simon Says*

El primer juego desarrollado para esta plataforma está inspirado en el juego *Simon Says* [23], y por ello se ha llamado de la misma. En este juego, las baldosas muestran una secuencia de cuatro elementos únicos que el usuario debe repetir en orden tras acabar de visualizar la secuencia completa.

En las figuras 5.6a a 5.6g puede verse una secuencia de ejemplo. Primero las baldosas aparecen apagadas hasta que se pulsa el botón *Start* que las enciende para mostrar qué color corresponde a cada baldosa. Al pulsar el botón *Play* comienzan a mostrarse las secuencias que hay que repetir.

En este ejemplo, el usuario deberá pulsar en orden las baldosas azul, naranja, verde y violeta, que en la lógica del juego equivale al vector  $[0,1,2,3]$ . Cuando el vector de la secuencia mostrada coincide con el vector de baldosas pulsadas por el usuario, éste recibe un punto y se registra el tiempo total necesitado para visualizar la secuencia y repetirla.



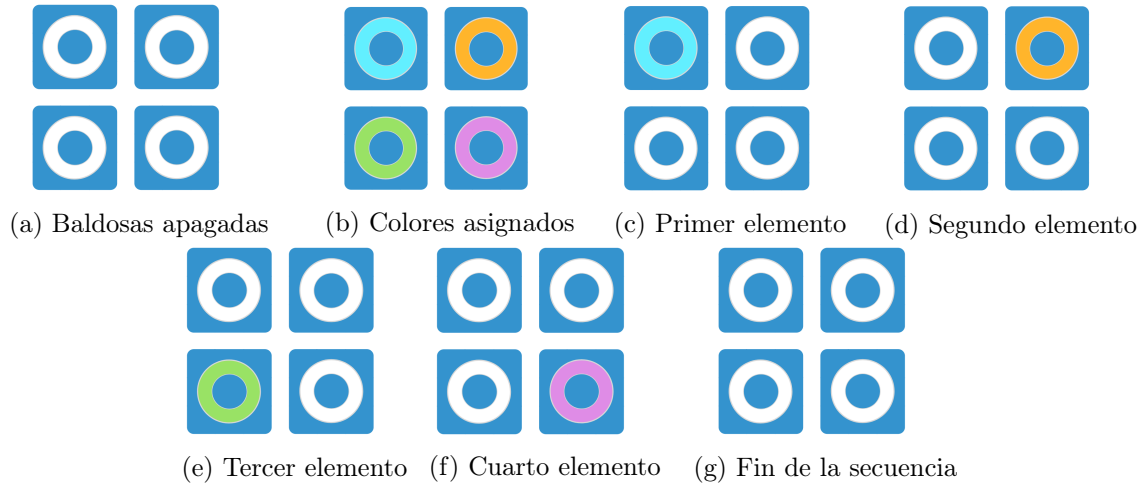


Figura 5.6: Secuencia de ejemplo del juego *Simon Says*

### Odd

El segundo juego desarrollado para esta plataforma llamado *Odd* funciona de manera similar al juego anterior. En este juego, las baldosas muestran una secuencia de cinco elementos. En este juego, a la vez que la baldosa se ilumina con su color, la aplicación también emite el color de manera verbal. En la secuencia existe un detalle especial ya que hay un elemento donde la aplicación emite el nombre del color pero no se ilumina ninguna baldosa. El usuario debe entonces identificar qué baldosa es la *intrusa* y pulsarla.

En las figuras 5.7a a 5.7h puede verse una secuencia de ejemplo. Primero las baldosas aparecen apagadas hasta que se pulsa el botón *Start* que las enciende para mostrar qué color corresponde a cada baldosa. Al pulsar el botón *Play* comienza a mostrarse secuencia de la cuál hay que identificar el elemento intruso.

En este ejemplo, el usuario deberá pulsar la baldosa naranja ya que es la que no se encendió al nombrar su color. Cuando el elemento intruso de la secuencia es identificado correctamente por el usuario, éste recibe un punto y se registra el tiempo total necesitado para visualizar y oír la secuencia, y pulsar la baldosa *intrusa*.

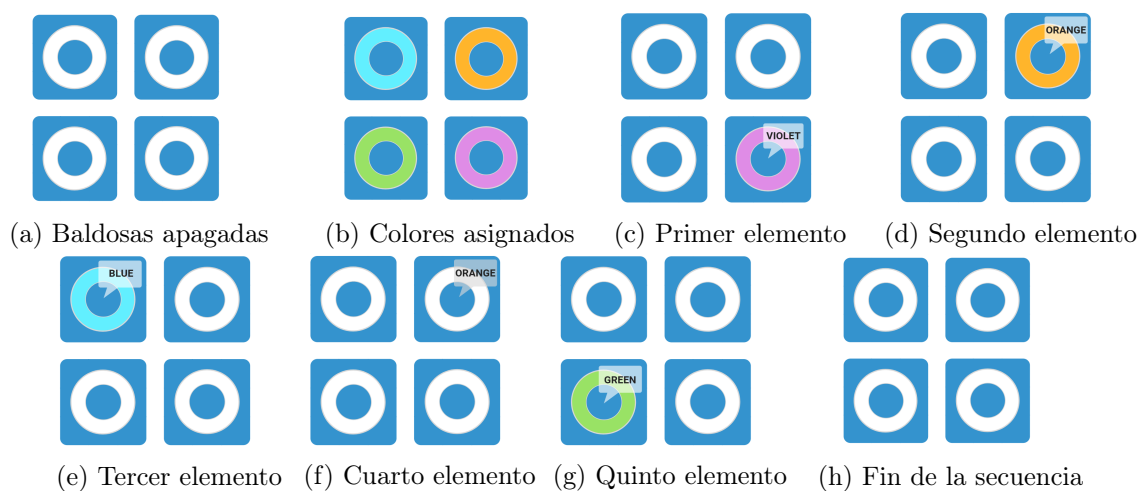


Figura 5.7: Secuencia de ejemplo del juego *Odd*

Este último juego es más sencillo pero pretende que la dinámica de juego sea muy rápida. El objetivo principal es pulsar la baldosa que se enciende lo más inmediatamente posible. Inicialmente se diseñó el juego para que fuera adaptativo, es decir, que fuera cambiando la velocidad a la que aparecían nuevos elementos según el porcentaje de aciertos del usuario. Sin embargo, esta funcionalidad pasó a ser manual (siendo necesario pulsar el botón *Play* cada vez que se desee que aparezca un elemento nuevo) por el motivo que se explica más adelante.

Cabe destacar que cada vez que se hace aparecer un elemento nuevo, éste puede ser cualquiera de las cuatro baldosas, pudiéndose repetir colores seguidos. Es decir, la lógica del juego implementada simula que se generan de manera aleatoria lo cual obliga a que el usuario siempre esté atento.

En las figuras 5.8a a 5.8d puede verse una secuencia de ejemplo. Primero las baldosas aparecen apagadas hasta que se pulsa el botón *Start* que las enciende para mostrar qué color corresponde a cada baldosa. Cada vez que se pulsa el botón *Play* se enciende una baldosa la cuál el usuario tiene que pulsar lo más rápido que pueda. En este ejemplo, el usuario deberá pulsar la baldosa azul.

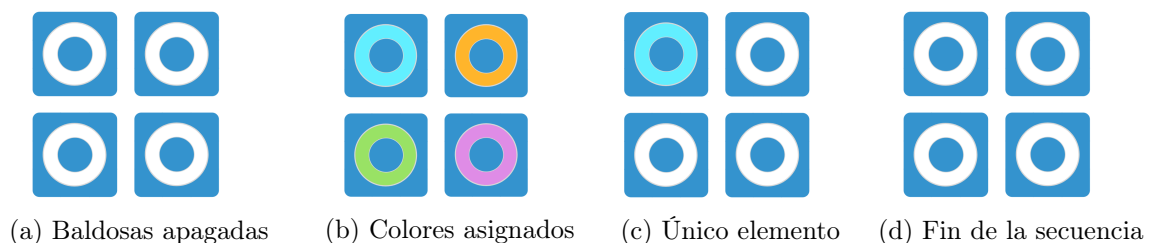


Figura 5.8: Secuencia de ejemplo del juego *Hit First*

Tras varias pruebas con usuarios, en base a la recomendación de los terapeutas, se incluyeron los siguientes apuntes:

- Las baldosas permanecen encendidas, cada una con su color, durante dos segundos al inicio, para así ayudar al usuario a recordar la relación entre baldosa y su color.
- Cuando se muestra una secuencia, hay un retardo de un segundo entre elementos ya que se encontró que ésta era la velocidad idónea.
- Al encender una baldosa cuando se muestra una secuencia, ésta permanece encendida durante medio segundo antes de apagarse y pasar al siguiente elemento, dando tiempo suficiente para asimilar ese elemento antes de pasar al siguiente.
- Cuando el usuario pulsa una baldosa, ésta se ilumina con el color que tiene asignado durante 0,4 segundos pudiendo el usuario ver instantáneamente el resultado de su acción.

En cuanto a la interfaz de usuario que controla los juegos, tal y como se ideó en el diseño, se ha implementado de la forma más minimalista posible. Es simple pero sin embargo contiene todo lo necesario para realizar los ejercicios. De esta manera se consigue una interfaz intuitiva, facilitando la comprensión de la dinámica de juego tanto al usuario como al responsable de manejar la aplicación, si estos no fueran la misma persona. A continuación se desglosa en detalle la función de cada botón, además de mostrar la apariencia de cada ventana.

Al abrir la aplicación aparece el menú principal que puede verse en la figura 5.9. Este menú incluye únicamente tres grandes botones correspondientes a cada uno de los juegos, que al pulsar nos lleva directamente a la ventana de cada uno.

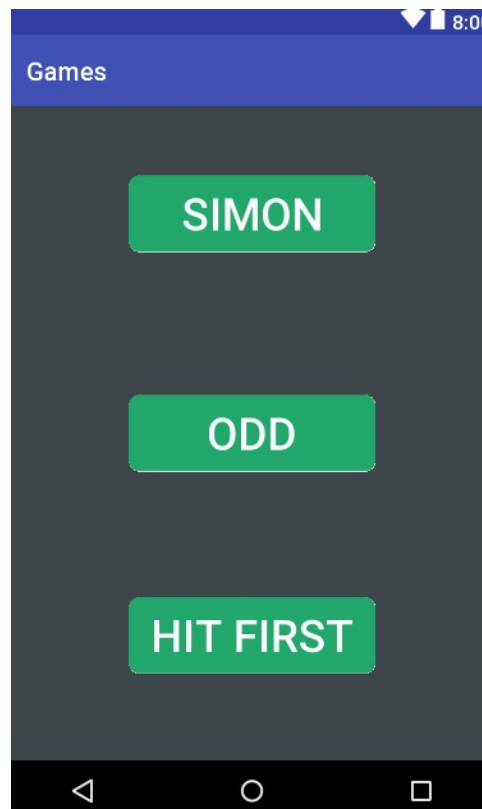


Figura 5.9: Menú principal de la aplicación

Una vez pulsas en cualquiera de los tres juegos, el usuario puede ver fácilmente cómo comenzar a jugar y cómo evolucionan los puntos. En las figuras 5.10a y 5.10b pueden verse dónde están posicionados cada botón en la interfaz de usuario de los juegos *Odd* y *Hit First*, respectivamente. La ventana principal del juego *Simon Says* se ha mantenido exactamente igual a la de *Hit First* para evitar confusión a los usuarios.

En el caso de que el usuario necesite ayuda para comprender el juego y cómo comenzar a jugar, se ha incluido un botón llamado *Rules* donde se explica con claridad qué pulsar para poner en marcha el juego y cuál es la dinámica del ejercicio. Esto junto con la imagen donde se muestra la disposición de las baldosas hace que sea sencillo comenzar.

El botón *Start* es un botón de tipo interruptor (o *toggle*) que es necesario pulsar para comenzar la sesión, y que si se vuelve a pulsar finaliza la sesión. Una vez pulsado *Start*, que inicializa las baldosas, se puede pulsar el botón *Play*.

El botón *Play* se debe pulsar cada vez que queramos jugar. Es decir, pulsarlo nos daría una única secuencia en el juego *Simon Says*, y hay que volver a pulsarlo para generar una secuencia nueva.

El hecho de que esta funcionalidad sea manual y no automática es debido a que se ha considerado importante que en una primera versión de esta plataforma sea el terapeuta quien decida cuál es la cantidad y velocidad idónea para cada persona.

Además, cada vez que el usuario realiza el objetivo del juego correctamente aumenta en un punto el contador grande (verde o azul). De esta manera se motiva al jugador a mejorar su rendimiento y, con ello, cumplimos nuestra meta de mejorar sus habilidades.

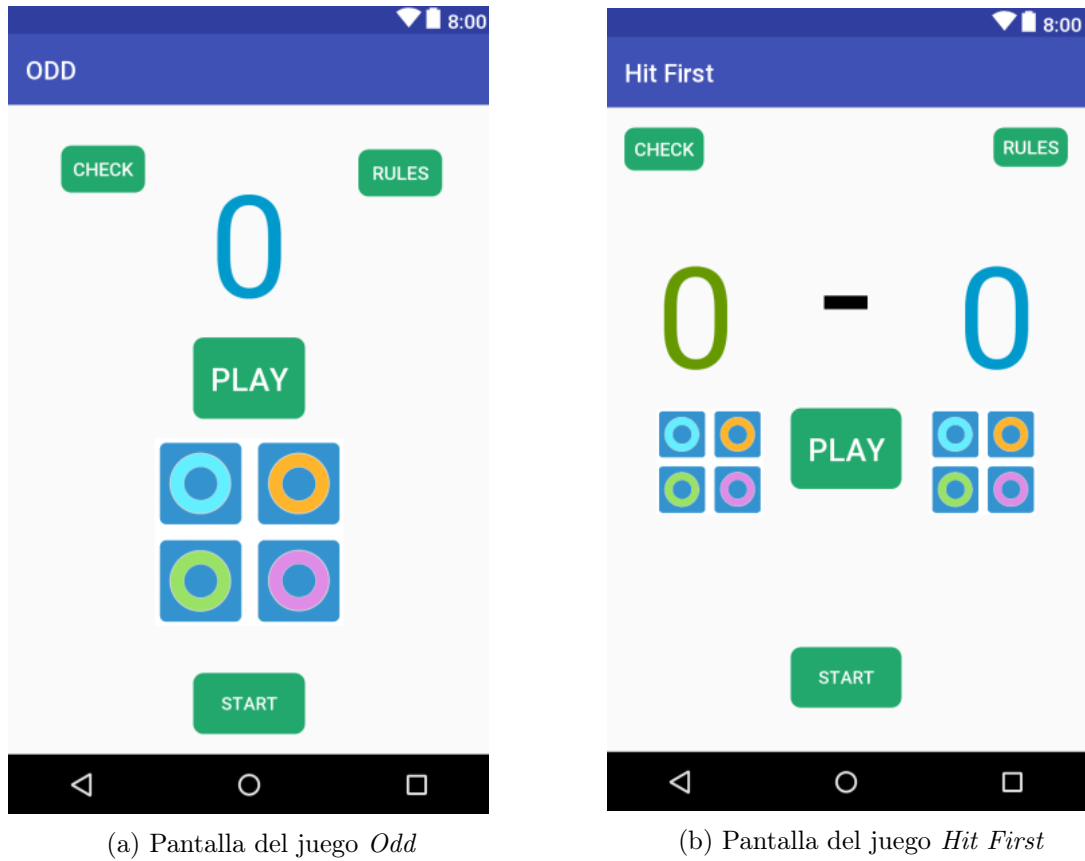


Figura 5.10: Interfaz de usuario de los juegos

Por último, el botón *Check*, que puede pulsarse en cualquier momento, muestra todos los tiempos obtenidos en una sesión. En la figura 5.11 se muestra un ejemplo de lo que aparece en pantalla al pulsar este botón. Son estos tiempos registrados los que se han guardado como resultados y luego analizado en la sección siguiente.



Figura 5.11: Ejemplo de datos de tiempos





# 6

## Experimentos realizados y resultados

### 6.1. Experimentos realizados

---

Esta plataforma fue evaluada por un grupo de personas de la residencia de mayores *Peder Lykke Centret* [24] localizada en Copenhague, Dinamarca. El grupo que participó en los experimentos fue constituido por ocho voluntarios de la tercera edad de entre 76 y 92 años, con una media de edad de 84 años, y cuya movilidad variaba entre reducida y muy reducida.

Ellos, junto con su monitora de ejercicios evaluaron la plataforma completa, la usabilidad de la aplicación, la dificultad de los ejercicios y su experiencia general, ofreciendo comentarios y sugerencias de mejora durante y al finalizar las pruebas.

Para llevar a cabo una recogida de datos para cuantificar la eficacia del uso de esta plataforma se idearon cuatro ejercicios, de los cuales se pueden ver los resultados obtenidos y su análisis en la sección siguiente.

Los tres primeros ejercicios se realizaron de manera individual, probando cada juego de la aplicación (*Simon Says*, *Odd*, y *Hit First*) tres veces por usuario, haciendo un descanso de tres a cinco minutos entre sesión. El último ejercicio se compuso por la realización de diez sesiones rápidas, sin descanso entre ellas, del juego *Hit First* por dos equipos de cuatro personas compitiendo entre ellos.

Las siguientes imágenes muestran algunos usuarios de la plataforma durante la realización de las pruebas propuestas.



Figura 6.1: Usuarios utilizando la plataforma

## 6.2. Resultados y análisis

En esta sección se exponen los resultados obtenidos de la realización de las cuatro pruebas descritas en la sección anterior. Todos los datos son tiempos en segundos.

Primero, se muestran las tablas correspondientes a los dos primeros ejercicios. En ellos se llevaron a cabo tres sesiones individuales de los juegos *Simon Says* y *Odd*. En estas dos tablas, los números en cursiva representan que la tarea correspondiente no se realizó correctamente. Es decir, en el juego *Simon Says* no se pulsó la secuencia correcta, y en el juego *Odd* no se habría pulsado correctamente la baldosa *intrusa*.

SIMON SAYS [Individual]			
Game Sessions	1	2	3
User 1	<i>16,120</i>	12,327	10,183
User 2	16,903	11,673	11,529
User 3	19,844	<i>10,129</i>	10,893
User 4	<i>21,469</i>	18,325	15,835
User 5	<i>15,993</i>	<i>14,836</i>	15,139
User 6	14,239	16,125	13,176
User 7	<i>26,876</i>	22,256	21,955
User 8	18,654	14,673	<i>10,344</i>

Cuadro 6.1: Resultados de tiempos del juego *Simon Says*



ODD [Individual]			
Game Sessions	1	2	3
User 1	20,058	16,637	18,275
User 2	27,647	23,574	24,013
User 3	34,027	18,673	26,175
User 4	21,685	19,945	12,937
User 5	10,683	12,649	17,739
User 6	13,584	23,836	19,129
User 7	19,046	26,361	23,916
User 8	24,168	21,836	22,038

Cuadro 6.2: Resultados de tiempos del juego *Odd*

La siguiente tabla muestra los datos de la tercera prueba: Tres sesiones individuales del juego *Hit First*. A diferencia de las dos tablas anteriores, en este caso no hubo ninguna pulsación errónea, ya que está en la lógica del juego sólo recopilar datos si la pulsación fue realizada correctamente.

HIT FIRST [Individual]			
Game Sessions	1	2	3
User 1	4,3967	4,1783	4,1673
User 2	2,9734	2,8452	2,6439
User 3	4,1880	3,7352	3,8013
User 4	4,7156	4,5824	4,6296
User 5	3,3500	3,8537	3,6045
User 6	3,9933	3,6956	3,5036
User 7	3,7453	2,9155	3,0563
User 8	2,8563	2,9563	2,8156

Cuadro 6.3: Resultados de tiempos del juego *Hit First*

Las dos tablas a continuación exponen los datos del último ejercicio. Al ser diez sesiones se han separado en dos tablas con cinco sesiones cada una. Como se ha mencionado en la sección anterior, este ejercicio corresponde a una prueba por equipos del juego *Hit First*.

HIT FIRST [Teams] <i>Sessions 1 - 5</i>					
Sessions	1	2	3	4	5
Team 1	2,4845	1,9107	2,0695	1,5109	1,4889
Team 2	2,9312	2,0212	1,8742	1,5816	1,6444

Cuadro 6.4: Resultados de tiempos de las primeras cinco sesiones del juego *Hit First* realizado por equipos

HIT FIRST [Teams] <i>Sessions 6 - 10</i>					
Sessions	6	7	8	9	10
Team 1	1,5083	1,5424	1,3932	1,3524	1,2133
Team 2	1,3823	1,3653	1,4233	1,1921	1,2613

Cuadro 6.5: Resultados de tiempos de las cinco últimas sesiones del juego *Hit First* realizado por equipos

Ahora se pasa a analizar los datos obtenidos. Para comenzar, se muestran tres gráficas correspondientes a los resultados de los tres primeros ejercicios.

Observando las gráficas 6.2, 6.3 y 6.4, dónde pueden verse los tiempos obtenidos por cada usuario a lo largo de las tres sesiones de cada ejercicio, se ve que, salvo en 6.2 donde se observa una disminución de tiempos general, no es sencillo lanzar conclusiones sobre las posibles mejoras de la psicomotricidad en los usuarios. Es por ello que más adelante se muestran y comentan otras gráficas de mayor interés y claridad.

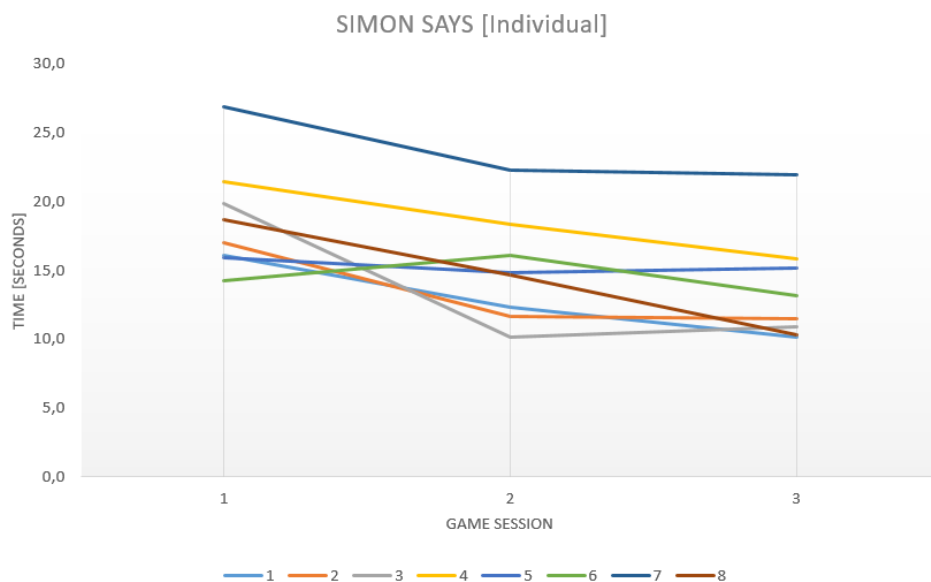


Figura 6.2: Gráfica con los resultados de cada usuario del juego *Simon Says*

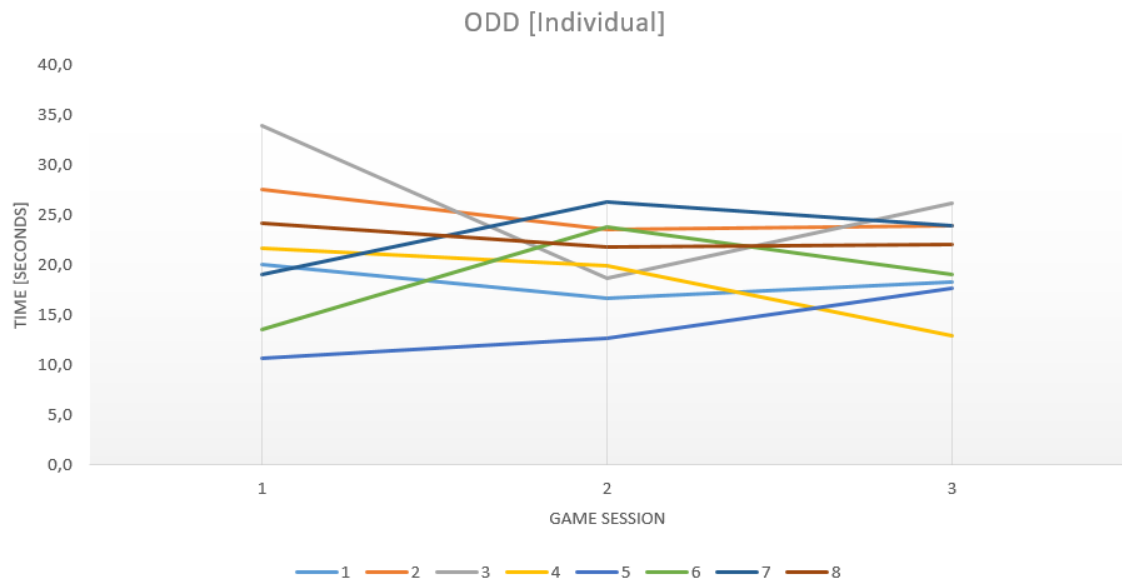


Figura 6.3: Gráfica con los resultados de cada usuario del juego *Odd*

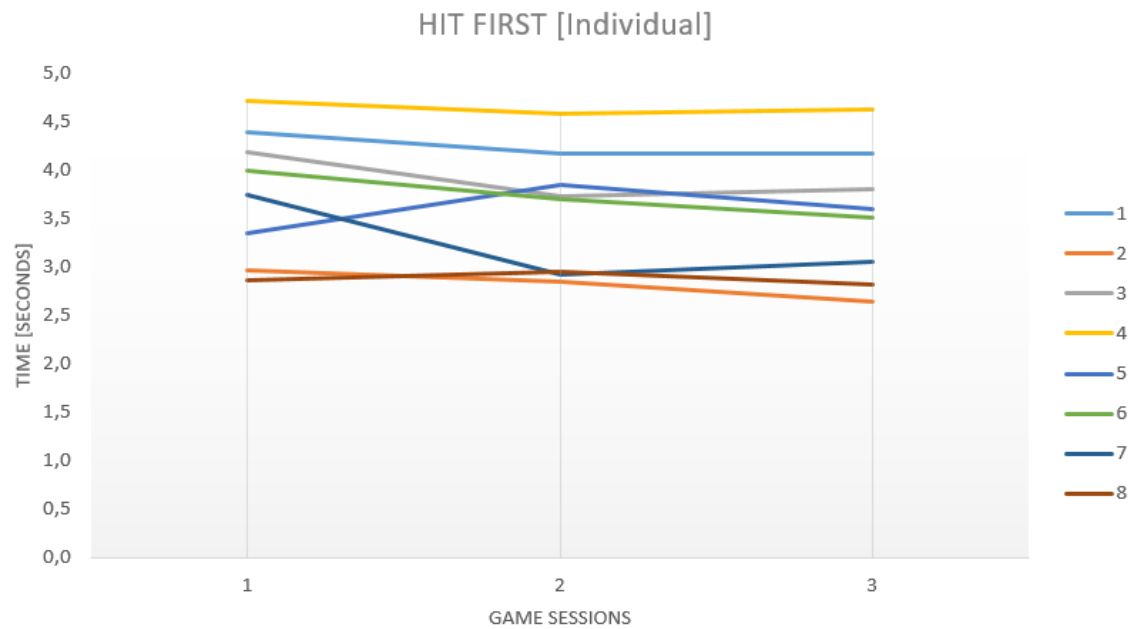


Figura 6.4: Gráfica con los resultados de cada usuario del juego *Hit First*

A continuación, la figura 6.5 muestra simultáneamente la media de tiempos obtenidos en cada una de las tres sesiones del juego *Simon Says*, así como el porcentaje de secuencias realizadas de manera incorrecta.

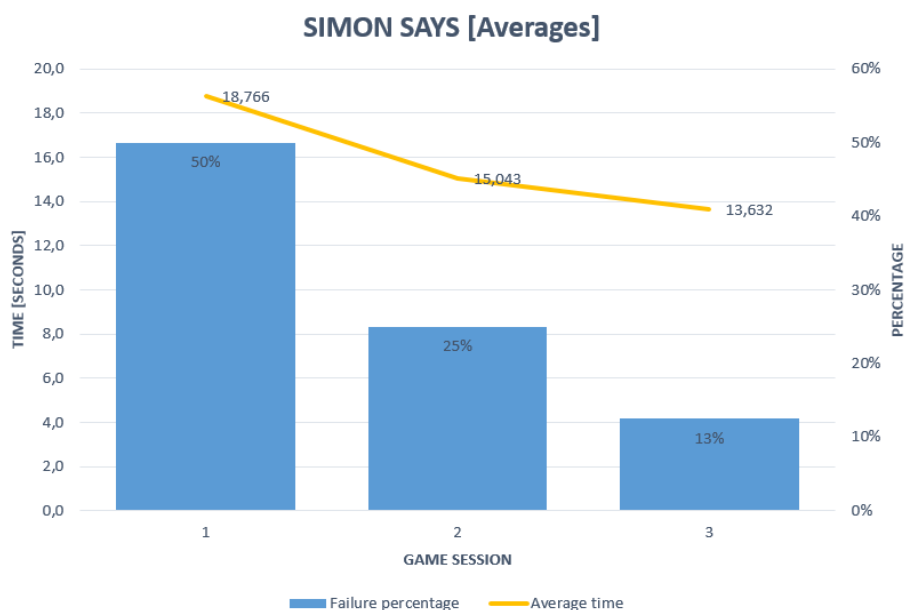


Figura 6.5: Gráfica con los resultados promedio de tiempo y porcentaje de fallos de las tres sesiones del juego *Simon Says*

Como puede observarse, tanto el tiempo de completar la secuencia como la cantidad de usuarios que fallan al realizar dicha secuencia disminuyen notablemente a mayor número de sesiones realizadas.

En concreto, en la primera sesión hubo un 50 % de usuarios que realizaron la secuencia de la sesión errónea con una media de tiempo de 18,755 segundos utilizados en completar el juego.

En el segundo intento del juego el porcentaje de fallos al realizar la secuencia se redujo a la mitad, siendo el 25 % de los usuarios los que se equivocaron. Así mismo, hubo una disminución del 20 % en cuanto al tiempo de ejecutar la secuencia respecto a la primera sesión, y la media de tiempo utilizado es de 15,043 segundos.

En la tercera sesión se produce una disminución del 9 % respecto a la segunda sesión en cuanto a tiempos, con una media de 13,632 segundos. Una diferencia menor que aquella de las primeras dos sesiones, pero que respecto a la primera sesión supone un total de 27,32 % en la disminución del tiempo necesario para realizar la secuencia del juego. En cuanto a errores cometidos, en la tercera sesión, únicamente un usuario comete algún fallo al jugar. Es decir, vuelve a disminuir a la mitad respecto a la sesión anterior.

Es interesante destacar que la mayor diferencia observable es entre la primera y la segunda sesión, ya que según los usuarios comprenden mejor el juego y se sumergen en él a medida que avanzan las sesiones, continúan las mejoras con un salto más sutil.

Estos resultados sugieren que este juego puede ayudar a mejorar el rendimiento del usuario, lo cual sugiere una mayor coordinación respecto a la psicomotricidad en relación tanto a la rapidez de movimientos como la agilidad mental de manera muy satisfactoria, ya que incluso con pocas sesiones se pueden ver unas mejoras muy notables.

La siguiente figura 6.6 muestra simultáneamente la media de tiempos obtenidos en cada una de las tres sesiones del juego *Odd*, así como el porcentaje de fallos cometidos por los usuarios en cada sesión.

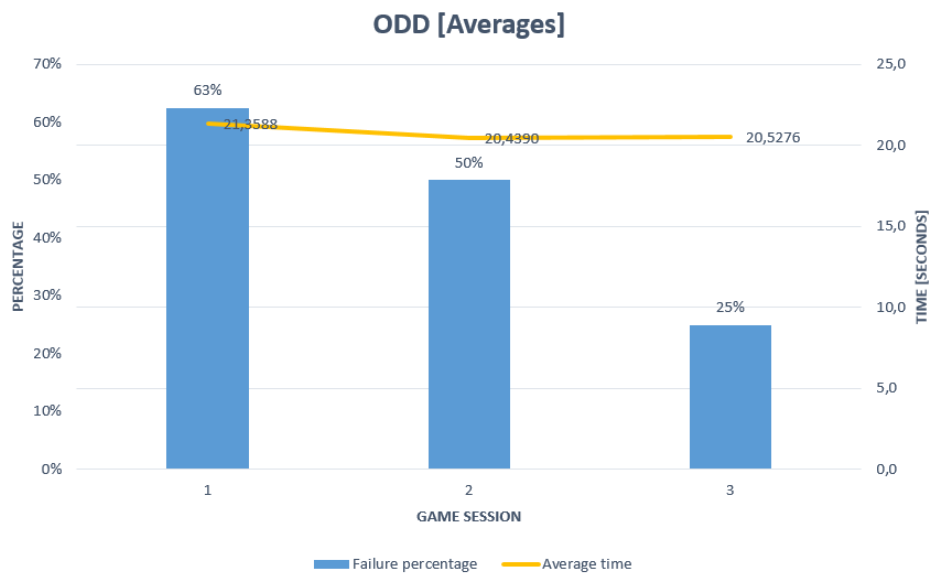


Figura 6.6: Gráfica con los resultados promedio de tiempo y porcentaje de fallos de las tres sesiones del juego *Odd*

Queda ilustrado que, aunque la cantidad de usuarios que fallan al realizar dicha secuencia disminuyen notablemente a mayor número de sesiones realizadas, el tiempo invertido de media no disminuye progresivamente.

En la primera sesión hubo un 63 % de usuarios que pulsaron la baldosa equivocada utilizando una media de tiempo de 21,358 segundos para observar, decidir y pulsar.

En la segunda sesión del juego, el porcentaje de errores al elegir qué baldosa pulsar se redujo hasta el 50 %. En cuanto al tiempo, hubo una disminución del 4,31 %, y la media de tiempo necesario es de 20,439 segundos.

En el tercer intento se produjo un ligero aumento del 0,43 % respecto a la segunda sesión en cuanto al tiempo utilizado, con una media de 20,537 segundos. Respecto a la primera sesión hay una disminución general de un 3,89 % del tiempo. En cuanto a fallos por parte de los usuarios, en la tercera sesión, se reduce a la mitad respecto a la sesión anterior.

Los datos anteriores analizados nos permiten ver que este juego resultó más complejo para los usuarios así como que la rapidez al pulsar la tecla no se mejoró de la manera esperada.

A continuación, la figura 6.7 muestra la media de tiempos obtenidos en cada una de las tres sesiones del juego *Hit First*.

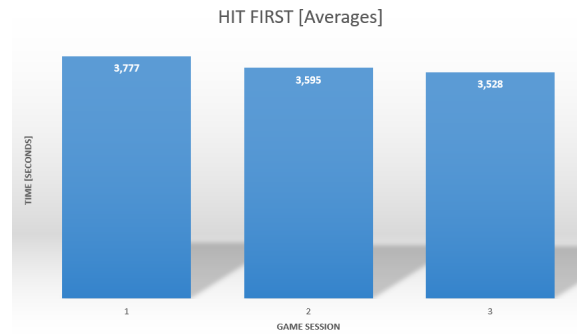


Figura 6.7: Gráfica con los resultados promedio de tiempo de las tres sesiones del juego *Hit First*

En esta gráfica se puede observar que el tiempo medio de pulsación de cada usuario disminuye a medida que avanza el número de sesiones.

Los tiempos medios de pulsación de la primera y segunda sesión fueron de 3,7773 y 3,5953 segundos, respectivamente. La diferencia de tiempos anterior supone una disminución del 4,82 % de la segunda sesión respecto a la primera.

La tercera sesión con una media de 3,5278 segundos de tiempo de pulsación muestra, sin embargo, una disminución más sutil respecto a la segunda sesión con una reducción de 1,88 %.

Como ocurre con el primer ejercicio, la mayor diferencia observable es entre la primera y la segunda sesión, ya que los usuarios entienden la dinámica del juego después de la primera toma de contacto.

Los resultados positivos de esta prueba podrían sugerir una mejora las habilidades psicomotrices de la manera intencionada.

A continuación, la figura 6.8 expone la media de tiempos obtenidos en cada una de las diez sesiones del juego *Hit First* realizadas enfrentando a dos equipos de cuatro.

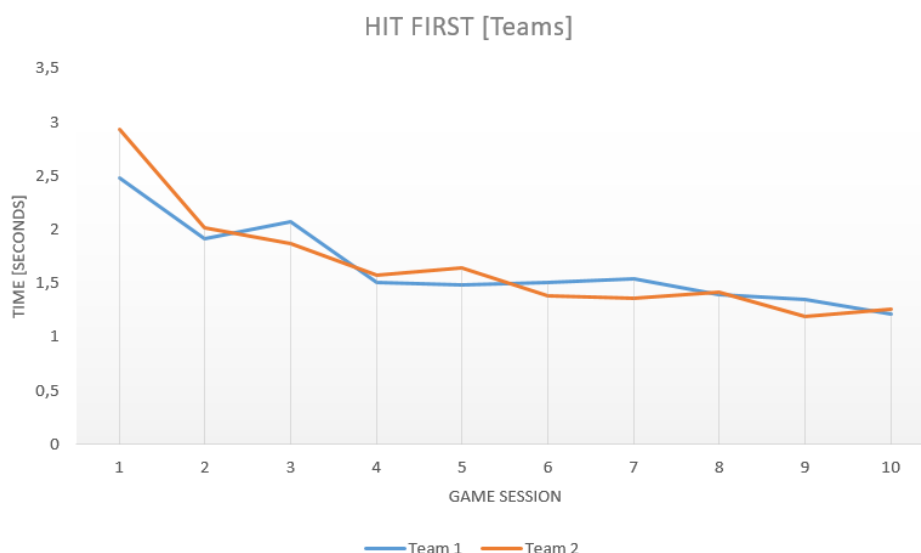


Figura 6.8: Gráfica con los resultados promedio de tiempo de las diez sesiones del juego *Simon Says* obtenidos por los dos equipos

---

En esta imagen se ve que el tiempo medio de pulsación de cada equipo disminuye a medida que avanza el número de sesiones, al igual que ocurre en el ejercicio individual de este mismo juego.

El dato más destacable es la disminución del 54,31 % del tiempo de pulsación de la última sesión respecto de la primera, a pesar de que entre sesiones la reducción del tiempo no es proporcional.

Este resultado nos muestra que este juego sugiere una posible mejora de las habilidades psicomotrices, y que además, se esperarían unos resultados aún mejores si se practicara con regularidad durante un tiempo más extenso.

A modo de conclusión es importante comentar las impresiones de los usuarios, tanto personas de la tercera edad como su terapeuta, acerca de cada juego. Éstas fueron recogidas mediante la realización de varias preguntas al terminar de llevar a cabo cada prueba y al finalizar la sesión completa.

El primer juego *Simon Says*, les resultó interesante y, en general, coincidieron en que la dificultad era adecuada, con una velocidad de muestra de la secuencia a repetir idónea. Además al tener que presionar todas las baldosas para realizar la secuencia del juego, les obligaba a moverse más que los otros dos juegos, y eso fue un detalle positivo comentado por parte de la monitora.

El segundo juego *Odd* resultó ser el más difícil de comprender y de realizar correctamente. Muchos lo encontraron ligeramente confuso y acababan pulsando la baldosa incorrecta, por lo que se obtuvieron los resultados descritos anteriormente.

El último juego, *Hit Firt*, resultó ser el que más gustó a los participantes, especialmente por la posibilidad de jugar en grupo. Esto les motivó a esforzarse en mejorar.

En general, los participantes constataron que al divertirse jugando les hacía olvidar que realmente estaban realizando ejercicios físicos y mentales, concentrándose únicamente en conseguir el objetivo del juego correctamente y lo más rápido posible. Además, la curva de aprendizaje inicial, tanto para las personas mayores con los juegos como para su monitora con la aplicación, fue adecuada y superable sin problemas.





# 7

## Conclusiones y trabajo futuro

Considerando el proyecto de forma global, esta plataforma para la mejora de las habilidades psicomotrices en personas mayores como Trabajo de Fin de Grado ha cumplido con los objetivos propuestos inicialmente. Se ha conseguido desarrollar e implementar satisfactoriamente, además de ser evaluada por usuarios a los que estaba dirigida esta plataforma, recibiendo comentarios positivos y sugerencias para un trabajo futuro.

Todo el proceso de diseño de la plataforma, la investigación del estudio del arte y de las herramientas necesarias, así como el proceso de diseño y desarrollo de la aplicación y juegos con mejoras constantes, ha requerido una inversión de tiempo mayor que la estipulada para un trabajo de fin de grado. Sin embargo, esto ha permitido a la estudiante añadir conocimientos útiles en el ámbito laboral, además de encontrar una motivación por realizar trabajos de ingeniería que aporten mejoras en la sociedad y en la calidad de vida de las personas.

Las siguientes mejoras que se podrían realizar en esta plataforma serían la de mejorar estéticamente la interfaz de usuario de la aplicación, y la de implementar una base de datos eficiente para la recolección de los tiempos de los usuarios, junto con un análisis de ellos ya incorporado. Además, acerca del controlador, se podría implementar en la aplicación la identificación de cada baldosa, evitando así tener que configurarlas manualmente en *C*.

Para finalizar, destacar la versatilidad de esta plataforma para la mejora de las habilidades psicomotrices. Esto abre camino a posibles futuras líneas de trabajo. Una sugerencia sería la de adaptar la plataforma a diferentes rangos de movilidad reducida de forma específica, así como la implementación de diferentes niveles de dificultad de los juegos. Otra línea de trabajo futuro podría ser la de incrementar el número de juegos diferentes, siguiendo sugerencias de profesionales del sector.



# Bibliografía

- [1] Henrik Hautop Lund. Playware abc: Engineering play for everybody. *Journal of Robotics Networks and Artificial Life*, 3(4):P21–P24, 2017.
- [2] Back-end, front-end. <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>. [Online].
- [3] Definición de psicomotricidad. <http://www.domusvi.es/articulo/la-psicomotricidad-en-mayores-que-es/>. [Online].
- [4] Habilidades que engloba la psicomotricidad. <https://emowe.com/psicomotricidad/>. [Online].
- [5] Ejercicios para la mejora de la psicomotricidad. <https://www.miterapiasinfronteras.com/psicomotricidad-adultos-mayores/>. [Online].
- [6] Stimulus. <https://stimuluspro.com/>. [Online].
- [7] Lumosity. <https://www.lumosity.com/>. [Online].
- [8] Henrik Hautop Lund. Playware research – methodological considerations. *Journal of Robotics Networks and Artificial Life*, 1(1):23–27, 2014.
- [9] Keil uvision 5 mdk. <http://www2.keil.com/mdk5>. [Online].
- [10] Arm microcontrollers. <https://www.arm.com/>. [Online].
- [11] C language. [https://www.tutorialspoint.com/cprogramming/c\\_overview.htm](https://www.tutorialspoint.com/cprogramming/c_overview.htm). [Online].
- [12] Android studio. <https://developer.android.com/about/>. [Online].
- [13] Java. <https://www.thoughtco.com/what-is-java-2034117>. [Online].
- [14] Xml. <https://www.w3schools.com/xml/>. [Online].
- [15] ANT. <https://www.thisisant.com/developer/ant-plus/ant-antplus-defined>. [Online].
- [16] Banda de frecuencia ism. <http://microondasism.blogspot.com/2012/10/bandaism-lasbandas-ism-industrial.html>. [Online].
- [17] Bluetooth. <https://www.bluetooth.com/>. [Online].
- [18] Leds rgb. <https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds?view=all>. [Online].
- [19] Nordic semiconductor *nRF51422*. <https://www.nordicsemi.com/eng/Products/ANT/nRF51422>. [Online].

- 
- [20] Conector j-link. <https://www.segger.com/products/debug-probes/j-link/>. [Online].
- [21] Master-slave architecture. [https://www.ou.nl/documents/40554/349790/IM0203\\_03.pdf](https://www.ou.nl/documents/40554/349790/IM0203_03.pdf). [Online].
- [22] Documentación para el desarrollo android. <https://developer.android.com/reference/>. [Online].
- [23] Simón dice. [https://es.wikipedia.org/wiki/Sim%C3%B3n\\_dice](https://es.wikipedia.org/wiki/Sim%C3%B3n_dice). [Online].
- [24] Peder lykke centret, plejecentre. <http://www.pederlykkecentret.dk/>. [Online].







## Controlador en Java

Listing A.1: Código fuente de Constants.java

```
1 package com.playware.app.games;
2
3
4 public class Constants {
5     public static final byte CMD_IDLE = 0x08;
6     public static final byte CMD_SET_COLOR = 0x02;
7     public static final byte CMD_FUNC_SPIN_ALL = 0x03;
8     public static final byte CMD_FUNC_COUNT_UP = 0x04;
9     public static final byte CMD_FUNC_COUNT_DOWN = 0x05;
10    public static final byte CMD_FUNC_SPIN_INDIVIDUAL = 0x06;
11    public static final byte CMD_FUNC_COUNT_UP_FADE = 0x07;
12    public static final byte CMD_FUNC_RECORDING = 0x09;
13    public static final byte CMD_SET_ALL = 0x10;
14
15    public static final byte EVENT_PRESS = 0x01,
16        EVENT_RELEASE = 0x09;
17
18    public static final byte LED_NUM_COLORS = 9;
19    public static final byte LED_COLOR_OFF = 0;
20    public static final byte LED_COLOR_RED = 1;
21    public static final byte LED_COLOR_BLUE = 2;
22    public static final byte LED_COLOR_GREEN = 3;
23    public static final byte LED_COLOR_INDIGO = 4;
24    public static final byte LED_COLOR_ORANGE = 5;
25    public static final byte LED_COLOR_WHITE = 6;
26    public static final byte LED_COLOR_VIOLET = 7;
27    public static final byte LED_COLOR_CYAN = 8;
28
29
30    public static byte getColourCmd(String colour) {
31        if (colour.equalsIgnoreCase("red")) return 0x01;
```

---

```
32     else if (colour.equalsIgnoreCase("blue"))    return 0x02;
33     else if (colour.equalsIgnoreCase("green"))    return 0x03;
34     else if (colour.equalsIgnoreCase("indigo"))    return 0x04;
35     else if (colour.equalsIgnoreCase("orange"))    return 0x05;
36     else if (colour.equalsIgnoreCase("white"))    return 0x06;
37     else if (colour.equalsIgnoreCase("violet"))    return 0x07;
38     else if (colour.equalsIgnoreCase("cyan"))    return 0x08;
39     else return 0x00;
40 }
41 }
```

---



```
1 package com.playware.app.games;
2
3 import java.util.Random;
4
5 import dk.dtu.antlibrary.AntConnection;
6 import dk.dtu.antlibrary.AntConnectionException;
7 import dk.dtu.antlibrary.AntData;
8
9 import static com.playware.app.games.Constants.*;
10 import static com.playware.app.games.MainActivity.tile;
11
12
13 public class TileCommander {
14     static byte messageId;
15     byte tileId;
16     byte color;
17     Random rand;
18     TileCommander(byte TileId) {
19         tileId = TileId;
20         color = LED_COLOR_RED;
21         rand = new Random();
22     }
23     public void setColour(byte colourCode) {
24         color = colourCode;
25         this.sendCommand(CMD_SET_COLOR, colourCode);
26     }
27
28     byte getColor() {
29         return color;
30     }
31
32     public void spinColours(byte colorId) {
33         color = colorId;
34         this.sendCommand(CMD_FUNC_SPIN_INDIVIDUAL, color);
35     }
36
37     public void setRandomColor() {
38         byte max = 4;
39         byte min = 1;
40         byte randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
41         while(randomNum == this.color) {
42             randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
43         }
44         this.setColour(randomNum);
45     }
46     public static int setRandomTile(int exceptTile, byte colorId) {
47         Random rand = new Random();
48         byte max = 3;
49         byte min = 0;
50         byte randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
```

---

```

51         while (randomNum == exceptTile) {
52             randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
53         }
54         tile[randomNum].setColour(colorId);
55         return randomNum;
56     }
57
58     public void setRandRedGreen(){
59         byte max = Constants.LED_NUM_COLORS;;
60         byte min = 1;
61         byte randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
62         if (randomNum % 2 == 0) {
63             randomNum = Constants.LED_COLOR_GREEN;
64         } else {
65             randomNum = Constants.LED_COLOR_RED;
66         }
67
68         this.setColour(randomNum);
69     };
70
71     public byte getRandomColor() {
72         byte max = Constants.LED_NUM_COLORS;
73         byte min = 1;
74         byte randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
75         while (randomNum == this.color) {
76             randomNum = (byte)(rand.nextInt((max - min) + 1) + min);
77         }
78         return randomNum;
79     }
80
81     public void spinColours() {
82         this.sendCommand(CMD_FUNC_SPIN_INDIVIDUAL, color);
83     }
84
85     void setColorId(byte colorId){ color = colorId;}
86     public void sendCommand(byte commandCode) {
87         AntData antData = new AntData(tileId);
88         messageId++;
89         antData.setBroadcastData(new byte[] {
90             tileId, messageId,
91             commandCode,
92             color, 0, 0, 0, 0
93         });
94         try {
95             AntConnection.getInstance().update(antData);
96         } catch (AntConnectionException e) {
97             e.printStackTrace();
98         }
99     }
100
101     public void sendCommand(byte commandCode, byte colourCode) {

```

---

---

```
102      AntData antData = new AntData(tileId);
103      messageId++;
104      antData.setBroadcastData(new byte[] {
105          tileId, messageId,
106          commandCode,
107          colourCode, 0, 0, 0, 0
108      });
109      try {
110          AntConnection.getInstance().update(antData);
111      } catch (AntConnectionException e) {
112          e.printStackTrace();
113      }
114  }
115 }
```

---

